which case step **778** deletes the keypress element without transmitting to the host and step **770** advances processing to the next element. Decision diamond **774** also invalidates the element if its press happened synchronously with other fingers of the same hand. Thus decision diamond **774** follows through on deletion command steps **601**, **612**, **615**, **620** of the synchronization detection process (FIG. **39**). Decision diamond **776** invalidates the keypress if too much lateral finger motion has occurred since touchdown, even if that lateral finger motion has not yet caused a chord slide to start. Because users may be touch typing on the surface, several millimeters of lateral motion are allowed to accommodate glancing fingertip motions which often occur when quickly reaching for keys. This is much more glancing tap motion than is tolerated by touchpads which employ a single finger slide for mouse cursor manipulation and a single finger tap for key or mouse button click emulation.

Decision diamond **780** checks whether the finger whose touchdown created the keypress element has since lifted off the surface. If so, decision diamond **782** checks whether it was lifted off soon enough to qualify as a normal key tap. If so, step **784** transmits the associated key symbol to the host and step **778** deletes it from the head of the queue. Note that a keypress is always deleted from the queue upon liftoff, but even though it may have stayed on the surface for a time exceeding the tap timeout, it may have still caused transmission as a modifier key, as an impulsive press with hand resting, or as a typematic press, as described below.

When a keypress is transmitted to the host it is advantageous for a sound generation device on the multi-touch surface apparatus or host computer to emit an audible click or beep as feedback to the user. Generation of audible click and beep feedback in response to keypresses is well known in commercial touchscreens, kiosks, appliance control panels and mechanical keyboards in which the keyswitch action is nearly silent and does not have a make force threshold which feels distinctive to the user. Feedback can also be provided as a light on the multi-touch surface apparatus which flashes each time a keypress is sent. Keypresses accompanied by modifier keypresses should cause longer flashes or tones to acknowledge that the key symbol includes modifiers.

If the finger has not yet lifted, decision diamond **786** checks whether its associated key region is a modifier such as <shift>, <ctrl>, or <alt>. If so, step **788** advances to the next element in the queue without deleting the head. Processing will continue at step **772** to see if the next element is a valid key tap. If the next element successfully reaches the transmission stage, step **784** will scan back toward the head of the queue for any modifier regions which are still pressed. Then step **784** can send the next element's key symbol along with the modifying symbols of any preceding modifier regions.

Decision diamond **782** requires that users touch the finger on the surface and lift back off within a few hundred milliseconds for a key to be sent. This liftoff timing requirement substitutes for the force activation threshold of mechanical keyswitches. Like the force threshold of mechanical keyswitches, the timing constraint provides a way for the user to rest the finger on the key surface without invoking a keypress. The synchronization detector **14** provides another way forefingers to rest on the surface without generating key symbols: they must touch down at the same time as at least one other finger. However, sometimes users will start resting by simultaneously placing the central fingertips on the surface, but then they follow asynchronously with the pinky a second later and the thumb a second after that. These latter presses are essentially asynchronous and will not be invalidated by the synchronization detector, but as long as they are not lifted within a couple hundred milliseconds, decision diamond **782** will delete them without transmission. But, while decision diamond **782** provides tolerance of asynchronous finger resting, its requirement that fingers quickly lift off, i.e., crisply tap, the surface to cause key generation makes it very difficult to keep most of the fingers resting on the surface to support the hands while tapping long sequences of symbols. This causes users to raise their hands off the surface and float them above the surface during fast typing sequences. This is acceptable typing posture except that the users arms will eventually tire if the user fails to rest the hands back on the surface between sequences.

To provide an alternative typing posture which does not encourage suspension of the hands above the surface, decision diamond **790** enables a second key acceptance mode which does not require quick finger liftoff after each press. Instead, the user must start with all five fingers of a hand resting on the surface. Then each time a finger is asynchronously raised off the surface and pressed on a key region, that key region will be transmitted regardless of subsequent liftoff timing. If the surface is hard such that fingertip proximity quickly saturates as force is applied, decision diamond **792** checks the impulsivity of the proximity profile for how quickly the finger proximity peaks. If the proximity profile increases to its peak very slowly over time, no key will be generated. This allows the user to gently set down a raised finger without generating a key in case the user lifts the finger with the intention of generating a key but then changes his mind. If the touch surface is compressible, decision diamond **792** can more directly infer finger force from the ratio of measured fingertip proximity to ellipse axis lengths. Then it can threshold the inferred force to distinguish deliberate key presses from gentle finger rests. Since when intending to generate a key the user will normally press down on the new key region quickly after lifting off the old key region, the impulsivity and force thresholds should increase with the time since the finger lifted off the surface.

Emulating typematic on a multi-touch surface presents special problems if finger resting force cannot be distinguished reliably from sustained holding force on a key region. In this case, the special touch timing sequence detected by the steps of FIG. **43**B supports reliable typematic emulation. Assuming decision diamond **798** finds that typematic has not started yet, decision diamond **794** checks whether the keypress queue element being processed represents the most recent finger touchdown on the surface. If any finger touchdowns have followed the touchdown represented by this element, typematic can never start from this queue element. Instead, decision diamond **796** checks whether the element's finger has been touching longer than the normal tap timeout. If the finger has been touching too long, step **778** should delete its keypress element because decision diamond **786** has determined it is not a modifier and decision diamond **794** has determined it can never start typematic. If decision diamond **794** determines that the keypress element does not represent the most recent touchdown, yet decision diamond **796** indicates the element has not exceeded the tap timeout, processing returns to step **770** to await either liftoff or timeout in a future sensor array scan. This allows finger taps to overlap in the sense that a new key region can be pressed by a finger before another finger lifts off the previous key region. However, either the press times or release times of such a pair of overlapping finger taps must be asynchronous to prevent the pair from being considered a chord tap.

Assuming the finger touchdown is the most recent, decision diamond **800** checks whether the finger has been touching for a typematic hold setup interval of between about half